

1. Consider the following instruction:

Instruction: **AND Rd, Rs, Rt**

Interpretation: **Reg[Rd] = Reg[Rs] AND Reg[Rt]**

(a) What are the values of control signals generated by the control in Figure 1 for the above instruction?

(b) Which resources (blocks) perform a useful function for this instruction?

(c) Which resources (blocks) produce outputs, but their outputs are not used for this instruction? Which resources produce no outputs for this instruction?

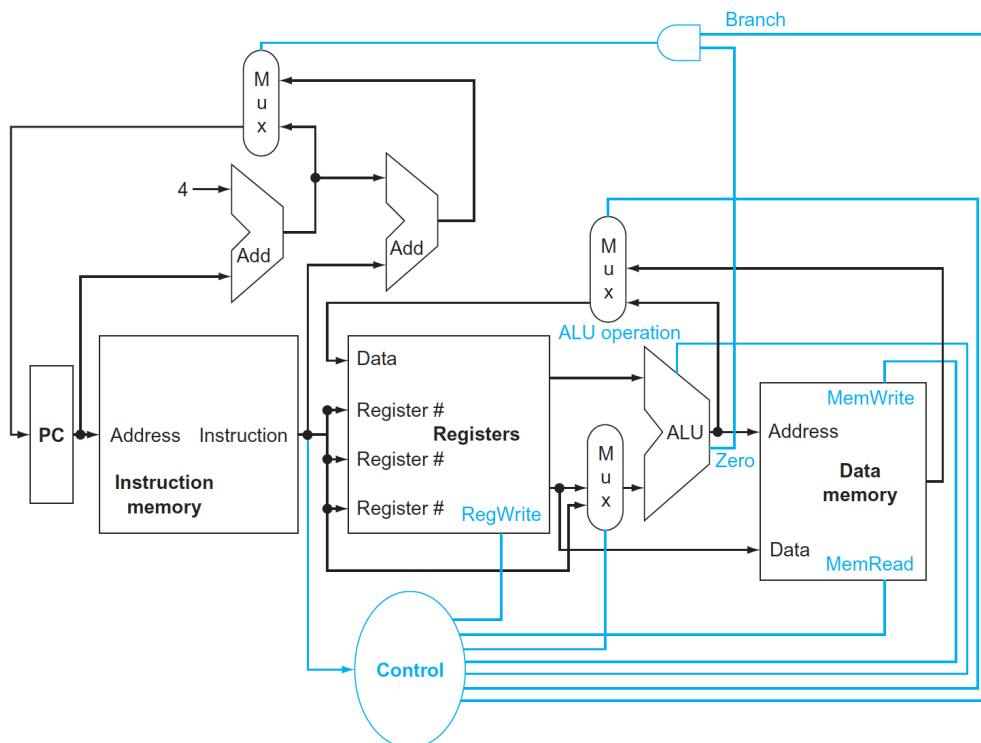


Figure 1 The basic implementation of the MIPS subset

2. The basic single-cycle MIPS implementation in Figure 1 can only implement some instructions. New instructions can be added to an existing Instruction Set Architecture (ISA), but the decision whether or not to do that depends, among other things, on the cost and complexity the proposed addition introduces into the processor data path and control. The first three problems in this exercise refer to the new instruction:

Instruction: **LWI Rt, Rd(Rs)**

Interpretation: **Reg[Rt] = Mem[Reg[Rd] + Reg[Rs]]**

(a) Which existing blocks (if any) can be used for this instruction?

(b) Which new functional blocks (if any) do we need for this instruction?

(c) What new signals do we need (if any) from the control unit to support this instruction?

3. For the problems in this exercise, assume that there are no pipeline stalls and that the breakdown of executed instructions is as follows:

add	addi	not	beq	lw	sw
20%	20%	0%	25%	25%	10%

- (a) In what fraction of all cycles is the data memory used?
- (b) In what fraction of all cycles is the input of the sign-extend circuit needed? What is this circuit doing in cycles in which its input is not needed?

4. In this exercise, we examine how pipelining affects the clock cycle time of the processor. Problems in this exercise assume that individual stages of the data path have the following latencies:

IF	ID	EX	MEM	WB
250ps	350ps	150ps	300ps	200ps

Also, assume that instructions executed by the processor are broken down as follows: ALU (40%), beq (10%), lw (30%), sw (20%).

- (a) What is the clock cycle time in a pipelined and non-pipelined processor?
- (b) What is the total latency of an LW instruction in a pipelined and non-pipelined processor?
- (c) If we can split one stage of the pipelined data path into two new stages, each with half the latency of the original stage, which stage would you split and what is the new clock cycle time of the processor?
- (d) Assuming there are no stalls or hazards, what is the utilization of the data memory?
- (e) Assuming there are no stalls or hazards, what is the utilization of the write-register port of the *Registers* unit?
- (f) Instead of a single-cycle organization, we can use a multi-cycle organization where each instruction takes multiple cycles but one instruction finishes before another is fetched. In this organization, an instruction only goes through stages it actually needs (e.g., ST only takes 4 cycles because it does not need the WB stage). Compare clock cycle times and execution times with single-cycle, multi-cycle, and pipelined organization.